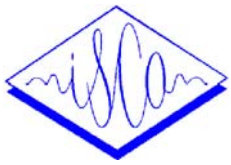

INTERSPEECH 2004 - ICSLP
8th International Conference on Spoken Language Processing
ICC Jeju, Jeju Island, Korea
October 4-8, 2004

Hybrid Model using Subspace Distribution Clustering Hidden Markov Models and Semi-Continuous Hidden Markov Models for Embedded Speech Recognizers

Youngkyu Cho, Sung-a Kim,and Dongsuk Yook

Department of Computer Science and Engineering
Korea University, Seoul, Korea
{ccameo,skim,yook}@voice.korea.ac.kr

Abstract

Today's state-of-the-art speech recognition systems typically use continuous density hidden Markov models with mixture of Gaussian distributions. Such speech recognition systems have problems; they require too much memory to run, and are too slow for large vocabulary applications. Two approaches are proposed for the design of compact acoustic models, namely, subspace distribution clustering hidden Markov models and semi-continuous hidden Markov models. However, these models require also large memory to acquire high recognition accuracy. In this paper, we propose a new hybrid model using subspace distribution clustering hidden Markov model and semi-continuous hidden Markov model with the aim of achieving much more compact acoustic models.

1. Introduction

The hidden Markov models (HMMs) has been widely used for automatic speech recognizers [1]. Each state of HMMs is modeled as a mixture of elementary probability density functions. To obtain higher recognition accuracy, HMMs typically require huge amounts of Gaussian distributions. Those models have a major impediment to their deployment in mass applications because they require too large memory, and are too slow to run. A significant challenge is to design these recognizers so that they may be run on more affordable machines of lower processing power and smaller memory size without losing accuracy [2]. One commonly used approach, which reduces both the computational cost and the size of models, is the use of parameter tying; for example, semi-continuous HMM (SCHMM) [4,5] and subspace distributions clustering HMM (SDCHMM) [2,3]. Both methods similarly divide the feature space into streams, and tie subspace (or stream) distributions across all states of all HMMs. Also, they can be derived from already existing continuous density hidden Markov model (CDHMM).

SCHMMs resemble the *M*-mixture continuous HMMs with all the continuous output probability density functions shared among all Markov states. Compared with the continuous mixture HMMs, the SCHMMs can maintain the modeling ability of large mixture probability density functions. In addition, the number of free parameters and the computational complexity can be reduced, because all the probability density functions are tied together, providing a good compromise between detailed acoustic modeling and trainability [6].

SDCHMM can represent original full-space distributions as some combinations of a small number of subspace distribution prototypes. Combinational effects of SDCHMM can be very powerful. For instance, a 3-subspace SDCHMM system with 128 prototypes can represent $128^3 = 2097152$ different full space distributions. SDCHMMs are also computationally efficient because if a small number of subspace Gaussians are shared by a large number of full space Gaussian components, log likelihood of these subspace Gaussians can be precomputed only once at the beginning of every frame, and their values can be stored in lookup tables [2].

In this paper, we propose hybrid models using subspace distribution hidden Markov models with powerful combinational effect and semi-continuous hidden Markov models with low computational complexity with the aim of achieving much more efficient acoustic models.

This paper is organized as follows. Section 2 describes the theory of the SDCHMM and SCHMMs, analyzes of SDCHMMs and SCHMMs the memory requirements. Section 3 introduces a new hybrid acoustic model using SDCHMM and SCHMM. Finally, we draw our conclusion in section 4.

2. Review of SDCHMMs and SCHMMs

In this section, we review the theory of subspace distribution clustering hidden Markov modeling and semi-continuous hidden Markov modeling. And we compare the memory requirements of SDCHMMs with those of SCHMMs.

2.1. Theory of SDCHMMs

The theory of SDCHMM is derived from already existing CDHMM. The observation probability of the *i*th state of a CDHMM is given by

$$P_i^{CDHMM}(O) = \sum_{m=1}^{M} c_{im} N\left(O; \mu_{im}, \sigma_{im}^2\right), \quad (1)$$

Condition & memory requirement	CDHMM	SDCHMM	SCHMM
#States pool	2000		
#Gaussians (39-dim)	20		
#codewords		256	
#subspaces or streams		39	4
Memory for codewords		79,872 byte	
Memory for indices		1,560,000 byte	
Memory for Gaussians	12,480,000 byte		
Memory for weights	160,000 byte	160,00 byte	8,192,000 byte
Memory for streamweigts			160,000 byte
Total memory requirement	12,640,000 byte	1,655,872 byte	8,431,872 byte

Table 1: Memory requirements of SDCHMMs and SCHMMs

where $P_i(O)$ is output probability of observation O at state i, c_{im} is the weight of the mth component of state i, μ_{im} and σ^2_{im} are the mean and variance of the mth component of state i [7]. To derive K-stream SDCHMMs from a set of CDHMMs, each Gaussian component is represented as a vector of K indices that indicate the particular set of subspace Gaussian components, which are to be used to calculate the likelihood. Thus,

$$P_i^{SDCHMM}(O) = \sum_{m=1}^{M} c_{im} \left(\prod_{k=1}^{K} N^{tied}\left(O_k; \mu_{imk}, \sigma^2_{imk}\right) \right), \qquad (2)$$

where O_k are the kth subspace of observation O, and μ_{imk} and σ^2_{imk} are mean and variance vectors of the mth mixture of ith state in the kth subspace, respectively. To derive K-stream SDCHMMs from a set of CDHMMs, the subspace Gaussians in each stream are clustered into a small set of L codewords. Each original subspace Gaussian is then approximated by its nearest subspace Gaussian codeword. For each Gaussian component in the system, therefore, it is only necessary to store the set of indices.

2.2. Theory of SCHMM

SCHMM may appear similar to SDCHMM. SCHMM also divides the feature space into streams, and ties stream distributions across all states of all HMMs.
The notations used in this paper are summarized as follows:

$P(O)$ state output probability given observation O;
O_k kth stream of observation O;
c_{imk} weight of the mth Gaussian of the ith state in the kth stream;
μ_{mk} mean vector of the mth Gaussian in the kth stream;
σ^2_{mk} variance vector of the mth Gaussian in the kth stream;
γ_k stream weight of the kth stream;
$N(\cdot)$ Gaussian pdf.

The observation probability of the ith state of a SCHMM is given by

$$P_i^{SCHMM}(O) = \prod_{k=1}^{K} \left(\sum_{m=1}^{M} c_{imk} N^{tied}\left(O_k; \mu_{mk}, \sigma^2_{mk}\right) \right)^{\gamma_k}. \qquad (3)$$

Each state output distribution of SCHMM is defined by M mixture component weight c_{imk} and the stream weight γ_k from each of streams. SCHMMs estimate a mixture of Gaussian densities for each of the streams independently, and then combine them with stream weights, while SDCHMMs do not. Each Gaussians of a stream is associated with its own mixture component weight, whereas a mixture component weight is shared among all the K-subspace Gaussians of a SDCHMM state [2].

2.3. Memory requirements of SDCHMM and SCHMM

Memory requirements of SDCHMMs and SCHMMs are compared in *Table 1*. We assume that there are 2000 state pool, 20 Gaussians for each state, 512 codewords, 39 subspaces for SDCHMMs and 4 streams for SCHMMs.

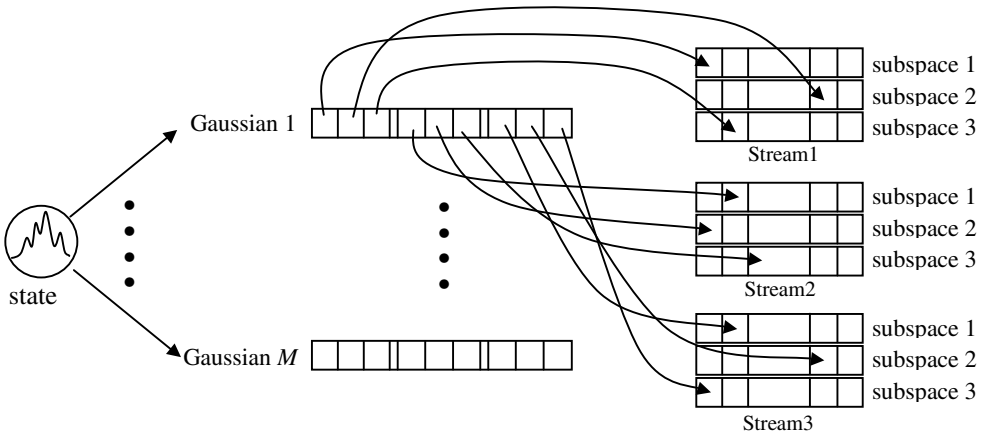

Fig. 1. A hybrid model with three streams and three subspaces per each stream

An implication of the difference in the scope of the assumptions is the number of streams required: The SCHMM favors fewer streams of higher dimensions, so that correlation among more features can be modeled and there will be fewer mixture component weights. Conversely, SDCHMM favors more streams of lower dimensions so that quantization of the subspace Gaussians of CDHMMs will give smaller quantization errors [2]. As can be seen in *Table 1*, Memory requirements of SDCHMMs and SCHMMs are reduced by 86% and by 34%, respectively.
Memory requirements for codewords, indices, weights, and stream weights are calculated as follows:

codewords =	#codewords × 39(dimensions) × 2(mean, variance) × 4bytes
indices =	#state pool × #Gaussians × #subspace × 1byte
weights	
for SDCHMM =	#state pool × #Gaussians × 4bytes
for SCHMM =	#state pool × #codewords × #streams × 4bytes
streamweights =	#state pool × #streams × 4bytes.

For each Gaussian component in the SDCHMM, it is necessary to store the index vectors. Each of stream Gaussians of SCHMMs is associated with its own mixture weight independently. Then, streams are combined with stream weights.

3. A Hybrid SDCHMMs and SCHMMs

In this section, we propose a new hybrid model using subspace distribution hidden Markov models with powerful combinational effect and semi-continuous hidden Markov models with low computational complexity.

3.1. Theory of hybrid model using SDCHMM and SCHMM

The theory of hybrid model is also derived from that of the CDHMMs, using the following notations:

$P(O)$ state output probability given observation O;

O_{sk} sth subspace for the kth stream of observation O;

c_{imk} weight in the ith sate of the mth mixture in the kth stream;

μ_{mk} mean vector of the sth subspace of the mth mixture in the kth stream;

σ^2_{mk} variance vector of the sth subspace of the mth mixture in the kth stream;

γ_k stream weight of the kth stream;

$N(\cdot)$ Gaussian pdf.

The observation probability density of state i is given by

$$P_i^{hybrid}(O) = \prod_{k=1}^{K}\left(\sum_{m=1}^{M} c_{imk}\prod_{s=1}^{S} N^{tied}\left(O_{sk};\mu_{smk},\sigma^2_{smk}\right)\right)^{\gamma_k}. \quad (4)$$

Formally, let us denote the full vector space of dimension D by R^D with an orthogonal basis. R^D is decomposed into K orthogonal streams R^{d_k} of dimension d_k. R^{d_k} is also decomposed into S orthogonal subspaces $R^{d_{k_s}}$ of dimension d_{k_s}.

Each of the original full space Gaussians is projected onto each of the S subspace Gaussians of dimension d_{k_s}.

Fig. 1 shows a scheme of new hybrid models. There are three streams with three subspaces per each stream in the example. In this scheme, we tie the stream Gaussians across all states of all SCHMMs. For each stream, we tie the subspace Gaussians across all SDCHMMs in each stream.

The equation of (4) may be implemented in three steps:

Step 1) First, train a one-stream Gaussian mixture model with *L* components onto the *K* streams and the *S* subspaces.
Step 2) Project all Gaussians in the original CDHMM onto those streams with subspaces.
Step 3) Tie the subspace Gaussians from all states and all phone models (CDHMMs) in each stream. This is done by clustering the subspace Gaussians into a small number of Gaussian prototypes in each subspace for each stream.

3.2. Comparison of memory requirements with other models

We compare our hybrid models with three other hidden Markov modeling methodologies: CDHMMs and SDCHMMs and SCHMMs. Memory requirements of hybrid model using SDCHMM and SCHMM are described in *Table 2*. We assume that there are 2000 state pool and 20 Gaussians for each state. For tied models, we assume that there are codebooks with 256 codewords and 4 streams with 3 subspaces per each stream.
Memory requirement for codewords, indices, weights, and stream weights are calculated as follows:

codewords = #codewords × 39(dimensions) × 2(mean, variance) × 4bytes
indices = #state pool × #Gaussians × #subspace × #streams × 1byte
weights = #state pool × #Gaussians × #streams × 4bytes
streamweights = #state pool × #streams × 4bytes.

Condition & memory requirement	A hybrid model
#States pool	2000
#Gaussians (39-dim)	20
#codewords	256
#subspaces (streams)	4 (3)
Memory for codewords	79,872 byte
Memory for indices	480,000 byte
Memory for Gaussians	
Memory for weights	640,000 byte
Memory for streamweigts	160,000 byte
Total memory requirement	1,359,872 byte

Table 2. Condition and memory requirement of Hybrid model.

As can be seen in *Table 2*, Memory requirement of hybrid model using SDCHMM and SCHMM are about 89% less than that of CDHMMM. Compared to SDCHMM and SCHMM, our hybrid models yield a more compact model set. Hybrid models using SDCHMM and SCHMM reduced memory requirements over 8%, compared with SDCHMM and SCHMM.

4. Conclusion

Compared with the continuous mixture HMMs, the SCHMMs and SDCHMMs can maintain the modeling ability of large mixture probability density functions. In addition, the computational complexity can be reduced, because all the probability density functions are tied together. Also SDCHMMs can simulate all original full-space distribution by some combinations of a small number of subspace distribution prototypes. SDCHMMs are also computationally efficient because log likelihoods of these subspace Gaussians can be precomputed only once at the beginning of every frame.

Therefore, our hybrid model using subspace distribution hidden Markov models and semi-continuous hidden Markov models may make acoustic models more compact, while they have powerful representational power.

5. References

[1] L. Gu and K. Rose, “Substate tying with combined parameter training and reduction in tied-mixture HMM design”, *Proceedings, IEEE Transactions on Speech And Audio Processing,* vol. 10, pp. 137-145, 2002.
[2] E. Bocchieri and B. Mak, "Subspace distribution clustering hidden Markov model", *IEEE Trans on speech and Audio Processing*, vol. 9, 2001, pp. 264-276.
[3] B. Mak, E. Bocchieri and E. Barnard, "Stream derivation and clustering schemes for subspace distribution clustering HMM”, *Proceedings, IEEE Automatic Speech Recognition Understanding Workshop,* pp. 339-346, 1997.
[4] J.R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition", *IEEE Transaction Acoustics, Speech and Signal Processing*, vol. 38, pp. 2033-2045, 1990.
[5] X. Huang, K-F. Lee, H.-W. Hon, "On semi-continuous hidden Markov modeling", *International Conference on Acoustics, Speech, and Signal Processing,* vol. 2, 1990, pp.689-692.
[6] X. Huang, A. Acero, H-W. Hon, *Spoken language processing*, Prentice-Hall, 2001.
[7] A. Aiyer, M.J.F. Gales, M.A. Picheny, "Rapid likelihood calculation of subspace clustered Gaussian components", *Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing,* vol. 3, pp.1519-1522, 2000.